

The following listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1-46. (canceled)

47. (currently amended) A network interface, comprising: ~~The network interface of claim 46~~  
a programmable media access control having at least one input configured to receive incoming frame data from a network; and  
a buffer manager coupled to receive the frame data from the media access control,  
wherein the media access control is operable in a first state in which it passes the incoming frame data to the buffer manager and in a second state in which it does not assert the incoming frame data to the buffer manager, the media access control is programmable to perform at least one filtering operation on each packet of the incoming frame data to determine whether to accept the packet, the media access control is also programmable to perform at least one additional operation in response to the incoming frame data, and the additional operation is not a filtering operation to determine whether to accept packets of the incoming frame data, wherein performance of the additional operation includes performance of at least one predetermined action in response to at least one predetermined bit pattern in the incoming frame data.

48. (previously presented) The network interface of claim 47, wherein the predetermined action is transmission of packet data.

49. (previously presented) The network interface of claim 47, wherein the predetermined action is issuance of a wake-up interrupt to a host computer.

50. (currently amended) A network interface, comprising: ~~The network interface of claim 46~~

a programmable media access control having at least one input configured to receive incoming frame data from a network; and

a buffer manager coupled to receive the frame data from the media access control, wherein the media access control is operable in a first state in which it passes the incoming frame data to the buffer manager and in a second state in which it does not assert the incoming frame data to the buffer manager, the media access control is programmable to perform at least one filtering operation on each packet of the incoming frame data to determine whether to accept the packet, the media access control is also programmable to perform at least one additional operation in response to the incoming frame data, and the additional operation is not a filtering operation to determine whether to accept packets of the incoming frame data, wherein performance of the additional operation includes transmission of diagnostic information.

51. (currently amended) A network interface, comprising:

a media access control having at least one input configured to receive incoming frame data from a network, wherein the incoming frame data comprises at least one packet; and

a buffer manager coupled to receive the frame data from the media access control, wherein the media access control is operable in a first state in which it passes the incoming frame data to the buffer manager and in a second state in which it does not assert the incoming frame data to the buffer manager, and is configured to perform a filtering operation on a destination address of each packet of the incoming frame data before asserting all of the packet to the buffer manager and to assert all of the packet to the buffer manager only if the filtering operation results in a determination to accept the packet.

52. (previously presented) The network interface of claim 51, wherein the media access control is configured to perform error checking on said each packet but only after the filtering operation results in a determination to accept the packet.

53. (currently amended) The network interface of claim 51, wherein the media access control is configured to ~~check~~ perform error checking for frame alignment errors, symbol errors, cyclic redundancy check errors, and length errors during performance of said error checking.

54. (currently amended) The network interface of claim 51, wherein the media access control is also programmable to perform at least one additional operation in response to the incoming frame data, and the additional operation is not a filtering operation to determine whether to accept packets of the incoming frame data.

55. (canceled)

56. (currently amended) ~~The method of claim 55, also including the step of:~~

A method for operating a network interface including a programmable media access control and a buffer manager coupled to the media access control, wherein the media access control is configured to receive incoming frame data from a network, the media access control is operable in a first state in which it passes the incoming frame data to the buffer manager and in a second state in which it does not assert the incoming frame data to the buffer manager, and the media access control is programmable to perform a filtering operation on each packet of the incoming frame data to determine whether to accept said packet, said method including the steps of:

programming the media access control to perform at least one additional operation in response to the incoming frame data, wherein the additional operation is not a filtering operation to determine whether to accept packets of the incoming frame data; and

operating the media access control to perform the additional operation, wherein said additional operation includes the step of performing at least one predetermined action in response to at least one predetermined bit pattern in the incoming frame data.

57. (previously presented) The method of claim 56, wherein the predetermined action is transmission of packet data.

58. (previously presented) The method of claim 56, wherein the predetermined action is issuance of a wake-up interrupt for a host computer.

59. (currently amended) ~~The method of claim 55, also including the step of:~~

A method for operating a network interface including a programmable media access control and a buffer manager coupled to the media access control, wherein the media access control is configured to receive incoming frame data from a network, the media access control is operable in a first state in which it passes the incoming frame data to the buffer manager and in a second state in which it does not assert the incoming frame data to the buffer manager, and the media access control is programmable to perform a filtering operation on each packet of the incoming frame data to determine whether to accept said packet, said method including the steps of:

programming the media access control to perform at least one additional operation in response to the incoming frame data, wherein the additional operation is not a filtering operation to determine whether to accept packets of the incoming frame data; and

operating the media access control to perform the additional operation, wherein said additional operation includes the step of transmitting diagnostic information.

60. (previously presented) A method for operating a network interface including a buffer manager and a media access control configured to receive incoming frame data from a network, wherein the media access control is operable in a first state in which it passes the incoming frame data to the buffer manager, and is operable in a second state in which it does not assert the incoming frame data to the buffer manager, said method including the steps of:

(a) operating the media access control to determine from a destination address of each packet of the incoming frame data whether to accept the packet, before asserting all of the packet to the buffer manager; and

(b) asserting all of the packet from the media access control to the buffer manager only when step (a) results in a determination to accept the packet.

61. (previously presented) The method of claim 60, also including the step of:

(c) operating the media access control to perform error checking on said each packet but only after step (a) results in a determination to accept the packet.

62. (previously presented) The method of claim 61, wherein step (c) includes the steps of checking for frame alignment errors, symbol errors, cyclic redundancy check errors, and length errors.